THE RAPID PROTOTYPE PROJECT

T. LeFebvre*, M. Mathewson, T. Hansen

NOAA Research−Forecast Systems Laboratory

Boulder, CO

Todd Dankers

National Weather Service, Boulder, CO

1. INTRODUCTION

Over the past three years, staff from the Forecast Systems Laboratory (FSL) have been working closely with National Weather Service (NWS) field forecast offices in an effort to evolve as quickly as possible the grid−editing component of the Interactive Forecast Preparation System (IFPS). This component called the Graphical Forecast Editor Suite (GFESuite) allows forecasters to define a weather forecast in gridded digital form. Once defined, the majority of NWS products are then derived from this digital forecast database.

Prior to IFPS, forecasters expressed most if not all of their forecasts in textual form. The paradigm shift from typing text products to expressing the forecast digitally represents one of the largest changes to forecasters in decades. At the project's outset, very little experimentation had been done with respect to verifying whether such and approach would be viable in a forecast office.

To reduce the risk in implementing a system with such a radically different paradigm, the NWS decided to deviate from its estabilshed software development, testing and delivery methodology and deliver software in a much more rapid fashon. That way, forecaster's comments could be more quickly utilized and the system could evolve in a much shorter time frame. This paper describes this rapid feeback/deliveryt process we call the Rapid Prototype Process.

2. HISTORY

The project began in 1992 with a major specifications document that described the high−level functionality of the system. Once the infrastructure was built, we invited forecasters from various regions of the NWS to travel to FSL, learn how to use the prototype system, and then freely comment on various aspects of the system including the user interface, ease of use/efficiency, meteorological soundness, and forecast methodology. These valuable forecaster comments were then incorporated into the prototype over the next 6−12 months, followed by another meeting where forecasters travelled again to repeat the exercise and refine the system further. This process continued for several years. We incorporated the valuable feedback from forecasters and the system evolved in a positive direction.

The rate of progress, however, was realtively slow. Because meetings were held relatively infrequently(6 − 12 months), coupled with the fact that the grid editing system was not used at all operationally, developers received little guidance from forecasters between meetings. The effort to integrate the grid−editing sytem with existing software destined for IFPS further inhibited progress toward an effective grid editor. The NWS Strategic Plan called for all NWS field forecast offices to be producing digital products by fall 2003. It became clear to many NWS managers that this goal would not be met if progress continued at current pace.

In June 1999, management representatives from each NWS region, NWS headquarters, as well as represetatives from the software development organizations met in Norman, OK ti dicuss the potential crisis of promising digital forecasts while lacking a system fully ready for operational use. At the meeting the attendees decided to implement a new experimental approach to software development. In this approach, FSL developers would release new software every 6−8 weeks instead of the 6−12 current done. Communication between forecasters and developers would take place via telephone conference calls, an e−mail bulletin board, and periodic face−to−face meetings. NWS managers and developers believed that this increase in communications would evolve the GFE more quickly from a prototype to an operational system. The name chosen for this experimental activity was the Rapid Prototype Project (RPP).

3. RAPID PROTOTYPE SOFTWARE

The RPP began with seven sites that would accept the

1 * Corresponding author's address: Tom LeFebvre, NOAA Forecast Systems Laboratory, 325 Broadway R/FS6, Boulder, CO, 80305;
e−mail: lefebvre@fsl.noaa.gov

prototype software, exercise it quasi−operationally, and provide feedback on a number of aspects. One site was chosen from each NWS region, and one from a national center, the Hydrometeorological Prediction Center (HPC) of the National Center for Environmental Prediction (NCEP).

An e−mail bulletin board (called the listserver) wes established that allowed anyone who registered to post questions and/or answers to technical problems. In addition, the listserver was frequently host to many philisophical discussions that ultimately refined the direction of new development.

In addition to the listserver, face−to−face meetings between forecasters and developers further enhanced information flow. Meetings typically featured activities such as training sessions to help forecasters understand how to use the newly−developed features and open discussions where new ideas were presented and refined. During these workshops, developers recorded these new ideas and suggestions for change so that they could be later incorporated in the GFESuite software.

3.1 RAPID SOFTWARE DEPLOYMENT

One of the critical ingredients to RPP was rapid software deployment. Approximately every 6−8 weeks, FSL developers delivered a new software release by mailing a CD−rom disk containing the software to each RPP site. To accomodate offices that were interested in receiving the these updates sooner, the software was also posted to a web−site for immediate download. Onced received, the local RPP focal point installed the software on a local computer for evaluation.

Delivering new software in this rapid fashion rather than waiting months between new releases offered several advantages. Forecasters continued to remain motivated to improve the system when they saw their suggestions implemented within weeks. As new software bugs were discovered by field forecasters, developers made patches available that fixed the problem, often within days. The Python programming language greatly helped in this manner, since it is a scripting language where no compiling or linking of the software is required. This feature of Python made field installation of patches as simple as installing a file in the approriate location and restart the software. Immediate fixes to problems in the field allowed forecasters to quickly continue their evalutaion work without having to wait weeks or months.

3.2 FORECASTER FEEDBACK

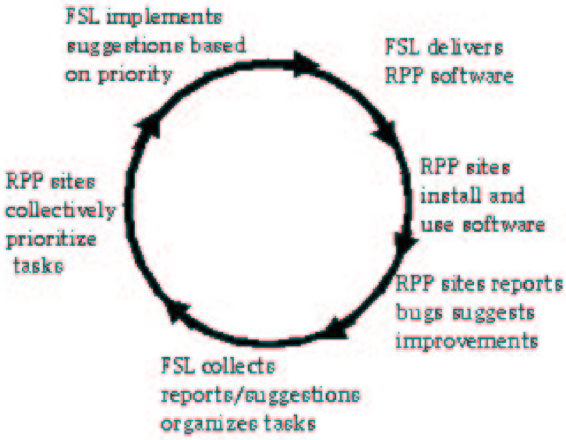
Rapid feedback from forecasters fuelled the speed with which new features were delievered. Armed with copius comments from the system users, developers confidently made changes consistent with forecaters wishes. As new reports of bugs or suggestions for new features were made available, developers carefully recorded and organized this feedback into a bug/change−request management system.

3.3 COMMUNICATION

In addition to the e−mail−based listserver communication, participants in the RPP met once per month via telephone conference call. At these calls each site typcially reported their status along with any problems they were currently experiencing. In addition, RPP members prioritized tasks derived from the listserver feedback collected previously factoring an estimated delievery date for each task. Each task was voted upon by each member. A priority derived from a consensus of the group was then recorded by the development team. This consensus priority was then use by developers to determine the order in which the tasks would be implemented.

Roughly ever 6−8 months, RPP field participants would meet the developers over a period of several days to discuss requirements, refinement of existing tools, new forecasting techniques, and brainstorm new ideas. These meetings typically began with a series of training exercises meant to educate each forecaster on new system features, tools, and user interface issues. Armed with this knowledge, developers and forecaters then discussed ways in which these new features could be improved in order to make the forecast process more efficient and intuitive. Open discussion sessions allowed forecasters to present and discuss new ideas that in many cases had never been discovered. As with the listserver and monthly conference calls, results of these sessions were recorded and converted into tasks. These new tasks were added to the master list of tasks and prioritized as before.

In general, each new software version focussed on a particular feature of the overall system. This limited the scope of new changes so forecasters could better concentrate their efforts on particular areas of the system and evaluate them.



4. HUMAN FACTORS

The set of activities that make up RPP not sufficient. Those who play the roles very important as well. Skills of individuals critical to success.

At least equal in weight (importance) with RPP process.

Development Team Characteristics
 − ability/willingness to wear all hats (requirements, design, coding, testing, support for customer −> satisfaction, high quality product.
 − while generally one member leads delvelopment, enough cross−pollination to ensure quality, more qualified staff for support.
 − A component for success is relatively high level of skill in both software engineering, science and meteorology. Important to communicate well with forecasters.
− Diverse mix of staff skills staff
 − big picture vs. details
 − no rigid enforcement of style, acceptance of operating styles. Visual (OO− design) vs. kinetic (code experimentation).
 − Leadership − willingness to listen to all ideas, also a developer

 − Many complex factors but we do know that this team works, but with only one data point it's difficult to pinpoint the precise reasons as with most synergistic systems. Recommend the group stay intact when the particular development project ends.

− True team approach
 − repect for all opinions
 − listen to ideas.

Forecaster Participants
 − motivation
 − much expertise, talent, key position,
 − satisfaction

Interaction between forecasters and developers
 − defines the motivation
 − the more responsive developers are, the more motivated forecasters are
 − Occasionally, rules may need be violated to foster parternship and motivation.

Team dynamics , internal (FSL) vs. external

Support as a priority − Help field forecasters

Partnership with forecasters

Motivation

Field Expertise

5. RESULTS

As RPP matured so did code. More sites wanted to participate. Grew to 17 volunteer sites, each with different area of expertise

6. CONCLUSION

RPP is a success, but must not be substituted for fully tested production style software. An excellent way to refine requirements, get testing from users, funnel feedback into next software version.

Because of the complex interaction of human factors, we recommend that the group remain intact to preserve the established working relationship with the group.

7. REFERENCES

Hansen, T. L., M. Mathewson, M. Romberg, 2000: Forecast Methodology using the GFESuite. Preprints, 17th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, Albuquerque, NM, Amer. Meteor. Soc.

LeFebvre, T. J., M. Mathewson, T. Hansen, M. Romberg, 2000: Injecting Meteorology into the GFESuite. Preprints, 17th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, Albuquerque, NM, Amer. Meteor. Soc.

[Nita − I need help with the reference to the Dec. 2000 FSL Forum articles.]